MINAMI

Micro-Nano integrated platform for transverse Ambient Intelligence applications

FP6 Contract: IST-034690

WP07 – Demonstration, validation & exploitation

Deliverable report

Deliverable ID:	**D7.5 v2**
Deliverable Title:	**Functional description of the last set of demonstrators (updated)**
Responsible partner:	TID
Contributors:	TID, NOKIA

Document Information

Document Name:	MINAmI_WP7_D7.5 v2
Document ID:	D7.5 v2
Revision:	Final
Revision Date:	13/11/2009
Author:	NOKIA
Security:	Confidential

Approvals

	Name	Company	Date	Visa
Technical Coordinator	Pascal ANCEY	ST Fr	13.11.2009	OK
WP leader	Miguel Angel Santiago	TID	21/09/2009	OK

Documents history

Revision	Date	Modification	Authors
v1.0	31/03/2008	Final version of the document	NOKIA
V1.1	22/07/2009	Comments to include test report	TID
V1.2	08/09/2009	Contribution from NOKIA, use case demonstrator, technology demonstrator and Test Report	NOKIA, TID
V.2.0	21/09/2009	Final version of the document	NOKIA, TID
Final	13/11/2009	Validation	Pascal Ancey

Section 1 - Executive summary

1.1 Description of the deliverable content and purpose

This deliverable contains the functional specification of the MINAmI second phase demonstrators, to be presented after TM18. The only demonstrator that belongs to this group is T7.4 “Memory download from passive tag”.

As specified in D7.1 **[1**], the primary target of this demonstrator is to demonstrate high speed downloading of data from a passive tag to a battery powered reader device (mobile phone). Compared to existing RFID technologies, such as NFC, the novelty in this case is dramatically increased memory capacity in the tag and communication speed in the order of tens of megabits per second and potentially also increased operating range. In addition to the “memory download from tag” use case, this demonstrator will also cover a more advanced scenario where the user can create content with his mobile phone (e.g. take a picture) and share it by writing (uploading) the data into a tag.

In the initial plans included in D7.1, the intention was to make an integrated demonstrator where an RFID reader prototype would have been attached into a mobile phone using some standard wired interface like USB. However, afterwards it has become clear that this is not a feasible approach and a better alternative is to split the demonstration into two clearly separate parts. First, the usability side can be addressed using an off-the-shelf mobile handset, existing wireless technologies and some clever tricks to give the test user an illusion that the technology is actually integrated into the product. Secondly, the technological side part i.e. the new RFID air interface is demonstrated separately and it focuses clearly on the technical issues without having the need to adhere to “productization” constraints like miniaturization, for example.

The following part of this document is structured so that the usability part of the demonstrator is first discussed in Section 2. It is followed by a description of the technology demonstrator in Section 3. A test report is included in Section 4. Section 5 includes the user manual and installation guide.

Glossary

Acronym	Signification
BER	Bit Error Rate
BT	Bluetooth
CRC	Cyclic Redundancy Check
FPGA	Field Programmable Gate Array
I-UWB	Impulse Ultra Wide Band
NFC	Near Field Communication
RFID	Radio Frequency Identification
UART	Universal Asynchronous Receiver / Transmitter
USB	Universal Serial Bus
UWB	Ultra Wide Band

Content

Section 2 - T7.4 part 1, "Use case demonstrator"

2.1 Brief description

This section describes the "look 'n feel" part of the T7.4 demonstrator. The target is to make a very product-like demonstrator that can be used for example to evaluate the use cases with test users. From end-user viewpoint, it is important that the demo can emulate the same user experience that would be available in a real product that uses MINAmI technology. For example, the data transfer speed perceived by the user must be no less than what is the target setting in MINAmI WP3 (10+ Mbit/s). Also, the demo equipment must work smoothly so that when the user touches a tag, for example, some kind of visual, haptic or other feedback must occur immediately. The form factor of the demo equipment must be also taken into account, ideally the hardware (in this case, the phone) should appear unmodified to the user.

Note that the requirements listed above make it imprartical to build an integrated demonstrator that would utilize the new MINAmI RFID technology in such a way that it can be used from a mobile phone. This is one of the reasons for abandoning the original plan of making one integrated demonstrator.

This part of the T7.4 demonstrator is based on two main uses cases, namely "memory download from a passive tag" and "social media based on mass memory tags". The first use case features only unidirectional data transfer (from tag to phone). The latter use case is an extension of the first use case and it involves also writing data from phone to tag. These use cases call for two different kinds of demonstration arrangements that are specified in detail in the following two subsections.

2.2 Memory download from passive tag

2.2.1 Structural specification

An overview of this part of the demonstrator is shown in Figure 1. The main components that constitute the demonstrator are:

1) a mobile phone capable of reading RFID tags
2) one or more passive RFID tags
3) a poster that the tag(s) are attached to

The mobile phone is used to activate the memory download and also display the content to the user. In this case, the content is a movie trailer that is downloaded to the phone and displayed on the phone screen. The passive tag provides the user with an interface to the service and the service can be accessed using the 'touch' paradigm that is one of the key characteristics in MINAmI technologies.

The role of the poster is to draw the user's attention to the service and potentially also give some guidance on how to use the service. The RFID tag that actually provides the service is embedded in the poster so that it is not even visible to the user.

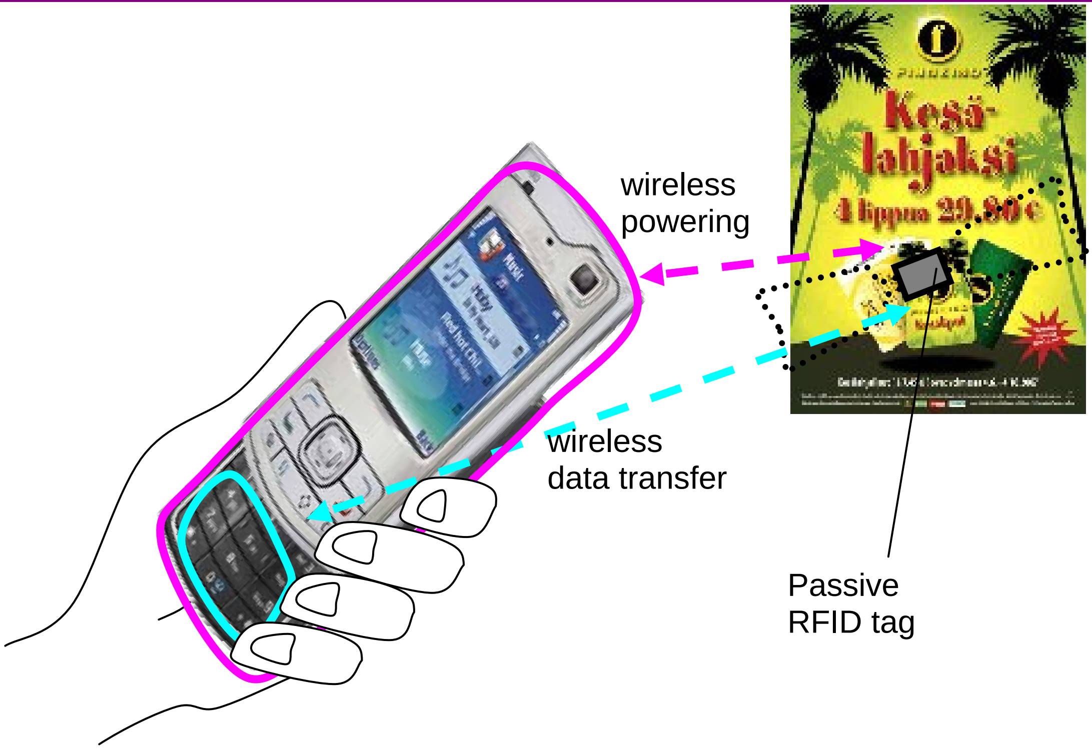

Figure 1 –Memory download from passive tag demo.

As can be seen from Figure 1, this is a very product-like demonstrator. For example, there is no external prototyping hardware attached to the phone and the tag is hidden on the backside of the poster. This is well inline with the requirements described earlier in this section.

2.2.2 Behavioral specification

The following excerpt from D7.1 is a very high level description of the scenario that this demonstrator is based on:

"*Mary (25 years) is relaxing in her sofa and reading a magazine covering the latest movies from Hollywood and elsewhere. After spotting an advertisement from a magazine she touches the page and a movie trailer and additional info how to book tickets are downloaded to her mobile. After pressing the start button the trailer is played in the mobile. The trailer will end with a possibility to order tickets.*"

The interactions that are included in the demonstration are as follows:

1) user decides to download a movie trailer from the poster to her phone
2) user activates the service by touching the tag with her phone
3) a menu is displayed on the phone screen, showing the movie title and a confirmation if the user wants to initiate download
4) once user accepts to initiate download, the data transfer begins
5) progress of the data transfer is indicated visually on the phone screen
6) when download is finished, phone prompts user to start playback

In step 5) listed above, the progress of the data transfer must be scaled so that it corresponds to the data rate that is expected as a result from the technology development in WP3.

2.2.3 Implementation

The demo implementation is based on NFC technology which is already available on the market. An NFC-enabled mobile phone (Nokia 6131 NFC) is used to emulate a MINAmI-compatible handset. NFC tags are available in a very convenient sticker form factor that are easily embedded in a poster, these are used in place of a MINAmI tag.

The interaction sequence discussed in previous subsection involves two technical challenges:

1) the service (movie clip download) must be activated by just touching the poster
2) the data must be transferred to the phone very fast (at least 10Mbit/s)

A solution to the first item is already available in NFC technology. The NFC standard includes a “Smart Poster” Record Type Definition that is tailored exactly for this kind of use cases. However, due to the limitations in memory capacity of NFC tags and NFC transfer speed (<1Mb/s), it is not possible to implement the data transfer with NFC technology.

A workaround to the data transfer challenge is to use a simple but effective conjuring trick as follows. Instead of actually transferring the movie clip from the tag to the phone, the movie is copied to the phone file system in advance. This means that there is no need to transfer any data from the tag to the phone, but the progress of the data transfer can be easily emulated by showing a progress bar on the phone screen so that the user is under an illusion that data is actually being transferred wirelessly.

Even if the current NFC tags cannot provide fast data transfer, they can still be used to transfer small amounts of data such as URLs or an SMS message. This NFC feature is utilized in this demonstrator and the following parameters (plain text) are stored into the tag:

- movie title
- name of the movie clip (identifies the file within phone file system)
- size of the file in bytes

The movie title is simply the name of the movie in plain text. When user touches the tag, the first thing displayed in the menu that prompts for further user actions is the movie name.

The second parameter is the name of the actual movie clip that is already preloaded into the phones internal memory. This parameter is used to link a tag into a file within the phone and it can be used for example to make a poster with multiple tags, each containing a reference to a different movie clip.

The file size is an essential parameter when emulating the data transfer speed. The demo application can check what is the size of the file to be transferred and divide it by the emulated data rate (e.g. 10Mbit/s) to calculate the total transfer delay in seconds. This information is then used to determine how fast the visual progress bar is updated and as a result, the data transfer delay perceived by the user is same as would be available using a real ‘MINAmI tag’. The close proximity of mobile phone to the tag is also monitored during the data transfer. If the mobile phone is pulled away more than few centimeters, the data transfer is stopped, and the user is informed to move the phone closer to the tag.

The equipment used in the demonstrator includes an Nokia 6131 NFC handset and one or more NFC tags that are available in sticker format. The hardware is not modified in any way.

The demo application in the phone is implemented as a Java MIDlet using the Nokia 6131 NFC Software Developers Kit (SDK) that is available for download at Nokia website. The implementation is based on standard protocols (NFCIP-1 **[2**] for RFID communication, 3GP format for video clips), thus there is no need to develop any proprietary protocols for this demonstrator .

2.3 Social media based on mass memory tags

2.3.1 Structural specification

An overview of this part of the use case demonstration is presented in Figure 2. The purpose of this setup is to enable a 'social media' use case so that the content in a mass memory tag can be dynamically updated by the users. For example, a user may take a picture with her camera phone and share it with other users by uploading the image file into the tag.

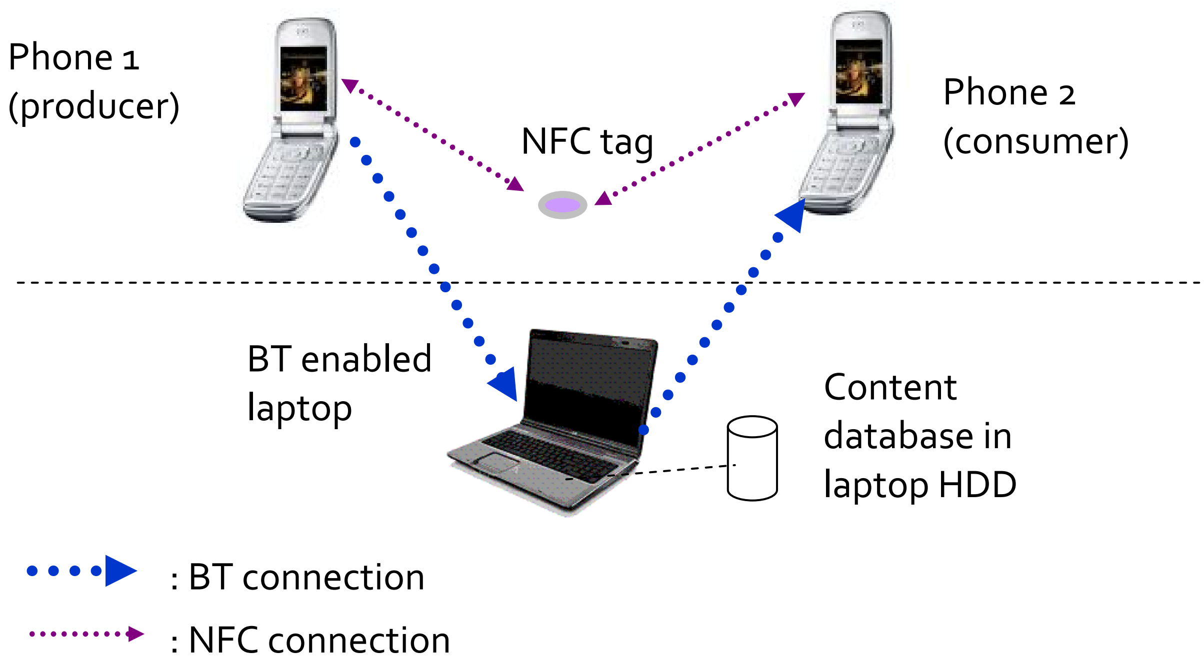

Figure 2 – Social media based on mass memory tags.

In Figure 2, the upper half that is above the dashed line is the part that is visible to the user. The main components in this part are:

1) a passive mass memory tag that provides both write and read access to shared content
2) phone 1 (or user 1), that generates some multimedia content and shares it with others by writing uploading it to the tag
3) phone 2 (or user 2), that reads the shared content from the tag

Phone 1 and phone 2 are hereafter referred to as '**producer phone**' and '**consumer phone**', respectively.

As there are no mass memory tags currently available, the demonstration setup includes some additional equipment that are needed to emulate the missing features. This part of the demonstrator is shown in below the dashed line in Figure 2. The main component here is a laptop that is connected via Bluetooth to both the *producer phone* and the *consumer phone*. The content database that includes all dynamic content produced by users is stored in the laptop hard disk drive. The *producer phone* can write content to the database though the Bluetooth connection. Similarly, the *consumer phone* can read from the database using Bluetooth.

2.3.2 Behavioral description

The key feature in this demonstration is the dynamic updating of the mass memory tag content. In the previous case (movie trailer download), the memory tag contained only static data to which the user only had read access. In this extended version, it is also possible for users to generate their own content and write it to the tag so that other users can access it.

From the *consumer phone* point of view, the behavior is basically identical to the 'movie trailer' case. The user accesses the tag in the same way as before, the only difference being that now the content is not fixed, but it can changed by other users.

In this demonstrator, the tag is used as a 'social media', comparable to for example the very popular YouTube portal that is accessible via internet. In order to provide better user experience to the *consumer phone* user, the following additional functionality should be included in the demonstration:

1) rating of content
2) dynamically generated "main menu" for the tag

In the first item listed above, the idea is that the user who downloads content from the tag can also provide some feedback and store it into the tag. A simple scenario would be that the user can rate movie clip stored on a tag using a scale from 1 to 5.

The purpose of the second item in the above list is to present the tag content to the user (*consumer phone*) in a user-friendly way. For example, if the tag contains lots of movie clips that are rated as described above, an obvious solution would be to present the tag content to the user in such a way that the top-rated movies are shown first. Another option would be to always present the most recent items first, regardless of their rating. In either case, the point is that the view that is shown to the user is not static, but it must be generated dynamically.

Note that the two new features listed above are not specified in detail here because they are relatively new ideas and it is still unclear what is the most suitable way to implement them from usability perspective. These issues should be further studied in WP1 (task T1.2 "Usage patterns and design requirements"), but it is not clear if there are enough resources available.

The interactions that are involved in writing content to the tag, i.e. the operation of the producer phone, are as follows:

1) user captures a video clip using the camera integrated into the phone
2) the video clip is stored in the phone memory (happens automatically)
3) user decides to share the clip by uploading it to the tag

4) user launches the sharing application and touches the tag to initiate upload
5) a menu is displayed on the phone screen, prompting the user to confirm if it is OK to upload data
6) once user accepts to initiate upload, she is prompted to touch the tag
7) when phone touches the tag, data transfer begins
8) progress of the data transfer is indicated visually on the phone screen
9) when data upload is finished, the phone returns to default mode

The sharing application that is launched in step 4) above can be implemented in different ways. In the simplest form, the application always uploads the most recent video clip file that has been captured with the phone camera. This kind of functionality is sufficient to demonstrate the use case. A more advanced sharing application could also offer possibility to select any video clip (or any file in general) in the phone memory to be uploaded into the tag. This would add more flexibility to the demonstration but it is not a mandatory requirement.

2.3.3 Implementation

The Bluetooth enabled laptop shown in Figure 2 plays a central role in the demonstration setup. It implements the functionality of the mass memory tag. The actual tag that is visible to the users is an NFC tag that is only used to trigger different activities in the application that is running on the phone and all data transfers are carried over the Bluetooth connections. The phone model used in the demonstration is Nokia NFC 6131 which includes both NFC and Bluetooth support.

To enable smooth operation, the Bluetooth connections are set up in advance before the demonstration is shown to the test users. Both the *producer phone* and the *consumer phone* are connected to the laptop using normal Bluetooth pairing. A server program is started on the laptop and it takes care of the following tasks:

1) allow the *producer phone* to write content to the database
2) generate a “main menu” that is shown for the *consumer phone*
3) allow the *consumer phone* to download content from the database

In item 2) above, the “main menu” refers to the information that is presented on the *consumer phone* screen when it touches the tag. This can be for example a list of filenames that are available in the tag. Additional information can be provided, too, but the main point is that the content is not static and therefore the “main menu” cannot be static, either.

For simplicity, the following assumptions/restrictions are made:
- only video clips are transferred to/from the tag
- video clips are in 3GP format (default video file format in the 6131 model)
- video clips are stored in the default location within phone file system (Gallery -> Video Clips)

The operation of the sharing application in *producer phone*, i.e. the steps involved in writing a video clip to the tag are listed below. It is assumed that the *consumer phone* user already has a video clip stored in the default location in her phone using the default naming convention (i.e. the file name is Video***.3gp, where *** is a running number).

1) When the sharing application is launched, it searches the latest video file from the ‘Gallery’ in phone memory, based on the file name (or optionally file timestamp)
2) A pop-up window is displayed to the user, showing the file name and asking permission to start upload

3) Once user select 'OK', the user is prompted to touch the tag with the phone
4) The application suspends and waits until it detects that an NFC tag has been touched
5) The selected file is transferred to the laptop over the pre-configured BT connection
6) A progress bar is shown to indicate user status of the simulated data transfer
7) When the simulated transfer delay is passed, the user is notified that transfer is complete
8) The actual file transfer over BT connection continues until it is completed

In step 6) in the above sequence, the duration of the simulated data transfer delay is determined similarly as in the movie trailer download demo (file size divided by assumed MINAmI tag transfer rate). Note that the actual data transfer over Bluetooth from phone to the laptop will take more time than the simulated delay, because the Bluetooth transfer rate is much lower than the simulated data rate. This means that the BT data transfer will continue even after the user has been notified that transfer was completed. The application must handle this in the background so that the user does not notice any additional delay.

As mentioned earlier, the operation of the *consumer phone* is basically the same as in the 'movie trailer download' case. The key difference is that instead of using a video clip that is pre-loaded into the phone memory, the video clip is now pre-loaded into the phone from the laptop over Bluetooth.

Because Bluetooth is not fast enough to emulate MINAmI RFID technology, it is necessary to start the data transfer from laptop to *consumer phone* before the user has even touched the tag. In practice, this can be implemented so that whenever the *producer phone* transfers a new file into the laptop, the server program running on the laptop must notify the *consumer phone* so that transferring of the new file to *consumer phone* begins immediately.

To ensure smooth operation, there must be enough delay between writing a file to the tag and reading the tag with another phone. Before the tag is read, the BT transfer that was initiated in the previous write operation must be completed. Assuming that the BT connection can reach 1 Mbit/s transfer rate, which is 10% of the nominal requirement in WP3 (10 Mbit/s), the ratio between real/simulated transfer speed is 1:10. In other words, the actual data transfer takes ten times longer than what is shown to the user.

A file size of 10 Mbit (1.25Mbytes) is quite reasonable for a small video clip with 176x144 resolution. Based on the assumptions made above, transferring this file from *producer phone* to laptop and from laptop to *consumer phone* would take 2 x 10Mbit / (1Mbit/s) = 20 seconds. However, from the users point of view (*producer phone*), the data transfer to tag appears to be complete in only one second. This must be taken into account in the demo arrangements.

Similarly as in the previous case, the demo application in the phone is implemented as a Java MIDlet using the Nokia 6131 NFC SDK. The RFID part is based on NFC. The communication between the mobile phones and the laptop is handled over Bluetooth. Some very lightweight proprietary protocol must be developed so that the server program running on the laptop can offer the basic services to the phones. The minimum set of services that need to be implemented is summarized below:

- upload file to the database
- get list of files in the database
- download file from the database

Section 3 - T7.4 part 2, “Technology demonstrator”

3.1 Brief description

This part of the T7.4 demonstrator concentrates on the technological aspects of the “mass memory download from tag” scenario. While the use case demonstrator described in previous section aims to emulate the user experience by using whatever technologies are currently on the market, the purpose of this part is to show that it is technically possible to implement a passive mass memory tag. As the usability demonstration is done separately, some of the requirements for this technology demonstrator are relaxed. For example, the demonstrator does not need to be miniaturized.

This technology demonstrator focuses on the high data-rate impulse UWB (I-UWB) air interface that is developed in WP3. The high-speed data transfer between a mobile reader and a tag is the key differentiating factor when comparing against existing RFID technologies like NFC. The planning and implementation of the demonstrator is done in tight co-operation between WP3 and WP7.

3.2 Structural specification

An overview of the demonstration setup is shown in Figure 3.

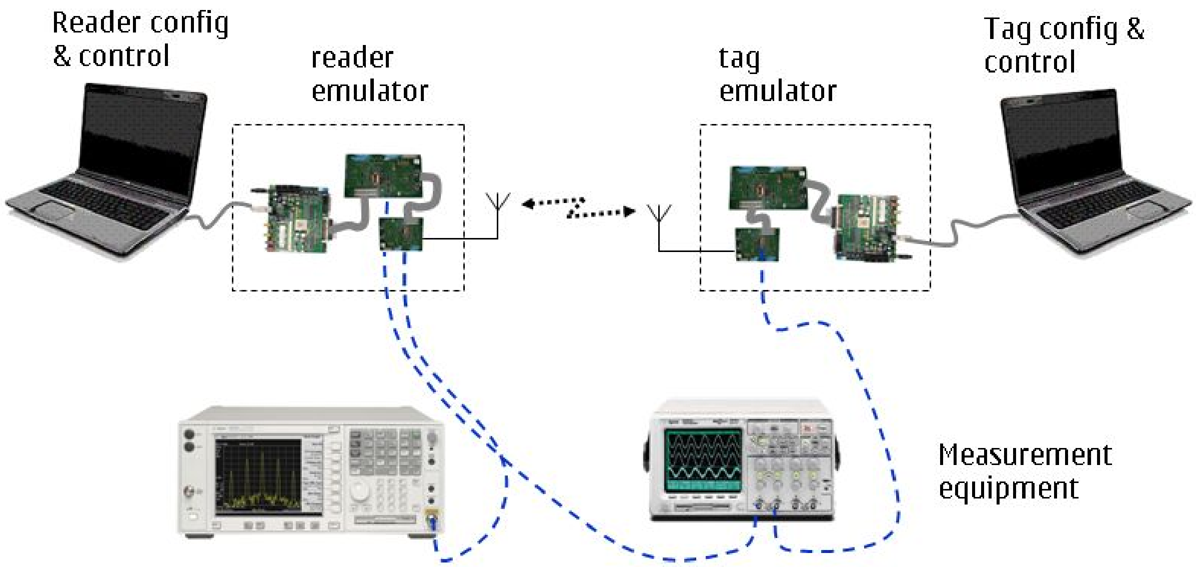

Figure 3 – Illustrative picture of the technology demonstrator.

The main components in the demonstration are the reader emulator and tag emulator. As shown in Figure 3, both of these are built from custom made test boards. For configuration and control, a laptop is attached to both the reader and the tag emulator. In addition to controlling the reader / tag hardware, the laptops can be also used for demonstration purposes such as visualization of data transfer progress. Measurement equipment such as spectrum analyzers or oscilloscopes can be also connected to the demonstrator.

3.3 Behavioural specification

This section summarizes the main objectives of the air interface demonstrator. At a very coarse level, it is essential to demonstrate the following two ‘features’:

1) interrogation: reader must detect that a tag is within its operating range and identify the tag
2) fast data transfer: target is 10 Mbit/s

These two main objectives are discussed in more detail below.

3.3.1 Interrogation

The term *interrogation* is used here to refer to the initialization of communication between a reader and a tag. During interrogation, a reader attempts to detect the tags that are within its operating range. A typical interrogation procedure consists of the following steps:

- Reader sends a pre-defined query command that is intended to reach all proximate tags
- The tags that respond to the query command, following a pre-defined protocol
- Reader detects and decodes the responses to identify the tags

The basic steps listed above are basically the same in existing NFC and EPCGlobal standards. An additional requirement that is characteristic to the I-UWB air interface that is under development in WP3 is the need for accurate timing synchronization between the reader and the tag. The maximum phase offset between reader and tag timing must be in the order of one nanosecond or less to enable successful data transmission.

The key challenge in the interrogation of I-UWB tags is the phase synchronization between reader and tag. Therefore, the first objective in this technology demonstrator is to show that a reader device can synchronize with an I-UWB tag with a precision that is sufficient to enable data transmission between reader and tag.

3.3.2 Data transfer between reader and tag

The second objective of the technology demonstrator is to show that the I-UWB air interface between the reader and tag meets the expectations in terms of maximum data rate. The target has been set to 10 megabits per second, which is significantly faster than any existing RFID technology today.

In order to show the fast data transmission capability, a large quantity of data needs to be transferred from the tag emulator to the reader emulator over the I-UWB air interface (see Figure 3). From technical point of view, the content of the data does not really matter, it is possible to for example use some pseudo-random binary sequence that is generated on the fly. The benefit of this approach is that the same sequence can be re-generated on the receiving end and used for error checking. This kind of setup makes it possible to do real-time measurements of bit error rate (BER), which is an important performance measure of the wireless link between the reader and the tag.

While pseudo-random data is suitable for testing the physical characteristics of the I-UWB link, it is not very suitable for demonstration purposes. Assuming that the data transfer will work as expected, the next phase in the demonstrator is to transfer some real multimedia data between the tag emulator and reader emulator to convince the audience that the

technology really works and to demonstrate the data transfer speed in a form that is understood by a non-technical audience.

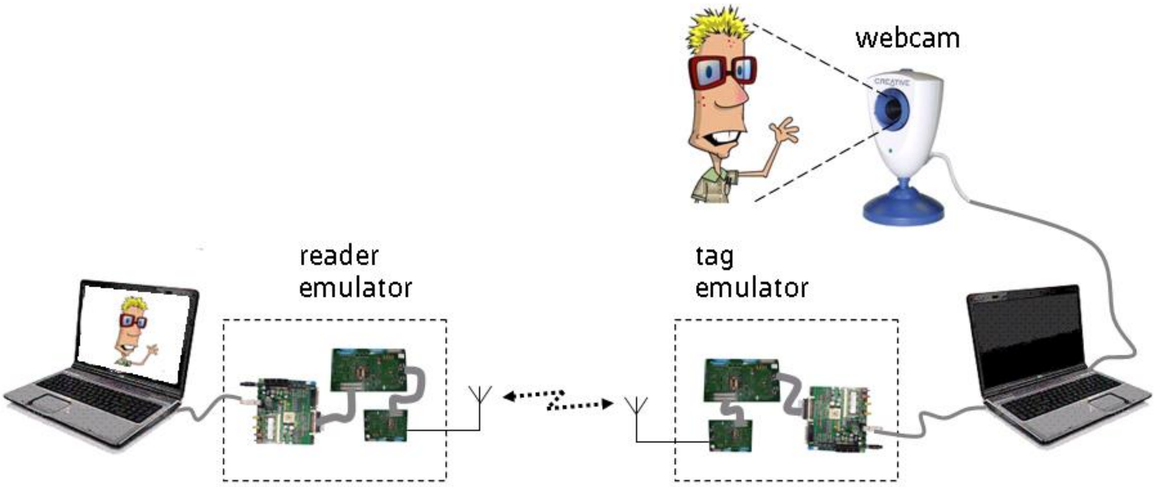

Figure 4 – Illustrative picture of an enhanced technology demonstrator.

Figure 4 illustrates an enhanced version of the technology demonstrator setup that was shown in Figure 3. For demonstration purposes, a digital camera is connected to the laptop that is attached to the tag emulator. Instead of using some random binary sequences, the data to be transmitted over the air interface can now be digital images captured by the camera and transferred to the tag emulator by the laptop. At the receiving side (reader), the received data is displayed graphically on the screen of the second laptop.

3.4 Implementation

3.4.1 Hardware setup

A block diagram of an I-UWB RFID transceiver is shown in Figure 5. The same setup is planned to be used to emulate both the reader and the tag functionality. Depending on the configuration data and SW image that is loaded to the FPGA device, the same hardware can act either as a reader or as a tag. The different components in Figure 5 are described below.

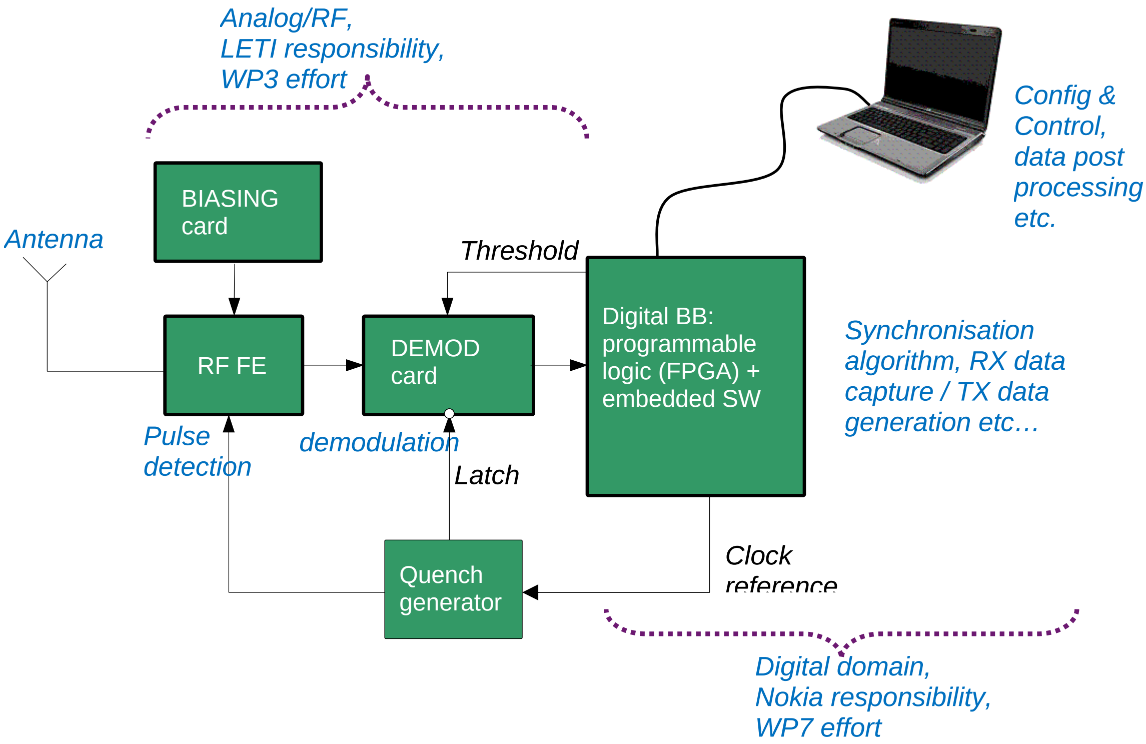

Figure 5 – Reader / tag emulator block diagram.

As shown in Figure 5, the hardware can be roughly divided into two parts: analog / RF side on the left hand side, and digital part in the right hand side. The same division is also used to split the responsibilities between WP3 and WP7.

The RF front end and analog part consists of multiple test cards that are tailored to perform some specific function of the transceiver. A biasing card is used to generate the bias current and voltages that are needed by the transceiver IC, which resides on another test card. The purpose of using a dedicated biasing card is mainly to avoid the need of large laboratory equipment and to make the demonstrator setup more portable.

The actual I-UWB transceiver chip is on a separate test card, marked as ‘RF FE’ in Figure 5. This part contains the super-regenerative I-UWB receiver.

The demodulation card demodulates received signal so that it can be passed in digital format to the digital part. An essential parameter that is required for successfully demodulate the received signal is the threshold, i.e. the amplitude level that represents the boundary between ‘0’ and ‘1’ symbols. This value is configured by the digital part, as shown in Figure 5.

The ‘quench generator’ board is used to generate the timing control signal (a.k.a. *quench*) required by the I-UWB receiver. The quench generator board can be used to control the timing of the I-UWB receiver with sub-nanosecond resolution.

The digital signal processing and control is included in one test card. The key part in the digital card is a large Altera Cyclone II FPGA device that can be configured to run either tag or reader algorithms. The following tasks that are done in the digital domain are common to both tag and reader modes:

- TX data encoding
- RX data decoding and error detection
- RX threshold level setting algorithm
- Timing control (keeping track of frames and slots)

3.4.2 Interrogation

As mentioned earlier, interrogation is one of the key elements in this technology demonstrator. The reader emulator must be able to detect the tag that is within its operating range and the reader and tag timing must be synchronized very accurately. The synchronization algorithm between reader and tag has not been finalized, but the key idea is as follows. The reader transmits interrogation signals (short impulses) and checks for responses from tag(s). The tag attempts to capture an interrogation signal from reader and if it does not detect one, it adjusts it's timing (phase) by a small step and tries again. This procedure is repeated until the reader and tag are in the same phase.

A central part in the interrogation procedure is the adjustment of timing at the tag side. The 'quench generator' board makes it possible to adjust the phase of the tag timing very precisely. Another essential part in the interrogation is the adjustment of the threshold setting for the demodulator. Both of these parameters (phase, threshold) can be configured by the FPGA that resides on the digital card. Using a very simplified description, the interrogation algorithm at the tag side works so that different phase and threshold settings are tested sequentially until an interrogation signal from reader is detected.

3.4.3 Data communication

After a successful interrogation, the reader and tag are in timing synchronization and it is possible to initiate actual data communication. The tag enters a "slave mode" so that it only tries to receive incoming commands from the reader, but does not initiate any activity on its own. The reader acts as a "master" in the communication, and it can either read data from the tag or write data to the tag by sending commands to the tag.

A minimal set of commands that is needed is one command for reading data (tag->reader), and one for writing data (reader->tag). The communication protocol between reader and tag is not yet defined, but an exemplary format for the 'read from tag' and 'write to tag' commands is shown in the two figures below.

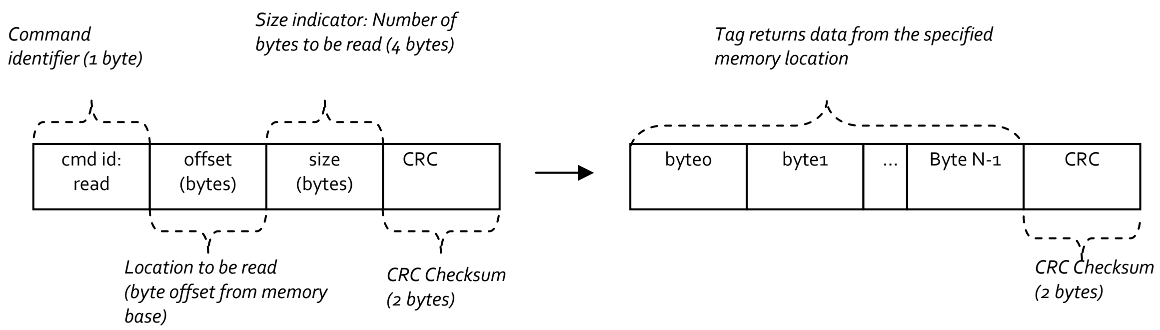

Figure 6 – Read command and response.

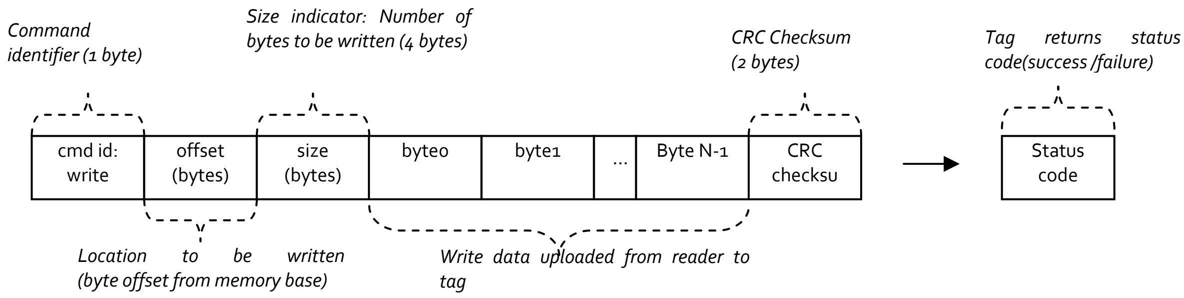

Figure 7 – Write command and response.

3.4.4 Interfaces towards the test laptop

As shown in Figure 5, the digital test card can be connected into a laptop. The digital board features an USB 2.0 controller that allows easy and fast connectivity between the laptop and the test board. Alternatively (or additionally), the digital board features also an UART so that the laptop can be connected through its serial port.

The connection between the laptop and the tag emulator can be used for various purposes, for example to initialize the memory of the tag using some multimedia content that is available on the laptop. This makes it possible to implement the enhanced demonstrator setup that was illustrated in Figure 4. At the reader side, the data that is received from the tag can be forwarded to the laptop for post processing. In case of the enhanced demonstrator shown in Figure 4, this would mean presenting the received image data on the laptop screen.

Section 4 - Test Report

This Test Report contains the testing activities done for *Memory tags demonstrator*.

The following documents have been used as reference for this annexe:

- D7.12: Final demonstrator requirements.
- D7.14: Test & Validation plan for the final set of demonstrators

4.1 Test Report

The following Test Report table contains the requirements and test cases that have been tested regarding the components involved in *Memory tags demonstrator*. The table structure contains the following information:

- **Requirement or Test Case TAG**: An identifier for the case tested.
- **Description**: A brief description of the case tested.
- **Status**: The status of the test. It can take the following values:
    - Blank: The case wasn't tested.
    - KO: The case showed an error when testing.
    - OK: The case had a correct result.
- **Comments**: Any commentary regarding the result of the test.

4.1.1 Requirements tested

The requirements to be tested appear in D7.12 *Final demonstrator requirements*.

Requirement TAG	Description	Status	Comments
NOK-UI-001	**Presentation of the tag content** The mobile phone shall be able to present the content stored in the tag memory.	OK	In movie trailer demo, the video clip is shown on the phone screen.
NOK-SW-001	**APIs used by mobile application** Software shall use standard APIs of the chosen mobile device.	OK	Movie trailer demo is implemented as Java MIDLet using NFC SDK provided by Nokia.
NOK-MP-001	**Presentation of the tag content** Mobile phone shall be capable of presenting the content of RF tag on its user-interface.	OK	In movie trailer demo, when user touches tag a menu appears that shows the name of the movie and instructions how to start playback.
NOK-MP-002	**Storing of downloaded data in phone** Mobile phone shall be able to store the content downloaded from the tag.	OK	
NOK-CI-001	**RF air interface for data and control** RF data and control transactions shall be performed in the UWB or 5GHz ISM band depending on the outcome of WP3.	OK	UWB is selected.
NOK-CI-002	**Remote powering method** Remote powering the tag shall be performed using 900MHz UHF ISM band.	(OK)	In the technology demo, it is not possible to fully

			power the tag from UHF, the integration level is not high enough. However, measurements and simulations indicate that in a fully integrated version (single chip), remote powering is feasible.
NOK-CI-003	**RF protocol** The reader and tag shall communicate using a low-power, low-range RF protocol. The protocol must support at minimum: - identification of tag - data transfer from tag to reader	OK	Both requirements are fulfilled and tested OK.
NOK-CI-005	**Bi-directional data transfer** Feasibility of bi-directional data transfer shall be studied.	OK	Data transfer works both ways, write-to-tag is also feasible (throughput is limited by available power)
NOK-CI-006	**USB host connection** Tag emulator and reader emulator shall be able to communicate with a PC or laptop via USB 2.0 connection at 10Mbit/s data rate.	OK	USB i/f works at >10M speed
NOK-FUN-001	**Test data generation** Tag/reader emulator shall be capable of generating a pseudo-random test pattern that follows some deterministic sequence. The same pattern can be generated at both the reader and the tag side.	OK	
NOK-FUN-002	**Error checking** Tag/reader emulator shall be capable of detecting transmission errors to determine BER performance. (Error correction is not needed)	OK	Tag is able to detect and keep statistics of bit errors.
NOK-FUN-003	**Phase synchronization** Tag and reader shall implement a synchronization algorithm that automatically aligns timing (phase) of the reader and tag so that I-UWB data transfer can take place. Synchronization must be fast enough so that no delay is perceived by the user.	OK	A simple but effective algorithm is used, synchronization delay is so small that there is no noticeable delay from user viewpoint.
NOK-PR-001	**Data transfer speed** The required download data-rate shall be at least 10 Mbit/s. The target is to increase bit-rate to 50 Mbit/s in the long run and therefore air-interface should have that potential to increase bit rate.	OK	56Mbit/s transfer rate is used.
NOK-PR-002	**Reading distance** Minimum read distance shall be at least 10cm preferably over 20cm. It must be also feasible to increase reading range in the future.		Verified by simulations and link budget calculations, but cannot be tested until fully integrated solution is available

			(beyond MINAmi scope)
NOK-TAG-001	**Power extraction in the tag** The tag shall be able to generate enough energy for wireless communication from the UHF signal transmitted by the reader that no external battery is needed.		See above.
NOK-TAG-002	**Tag interrogation** The tag shall be able to automatically initialize wireless communication and start transferring data when interrogator signal is activated by the reader.	OK	
NOK-PR-003	**Memory capacity of the tag** Memory capacity of the tag shall at least 64 Mbits. Feasibility of 1 Gbit memory capacity implementation shall be studied.		Memory capacity is limited only by available power. Not possible to test until fully integrated solution is available.
NOK-RB-001	**Robustness against transmission errors and powering failures** The memory content in the tag shall not corrupt upon data transfer or remote powering failures.	OK	All communication between reader and tag is protected by CRC.
NOK-HI-001	**HW interface between reader and phone** The reader component shall be connected to the mobile phone using selected high speed standard connector (e.g. USB).		This requirement is no longer valid, listed as <obsolete> in D7.12.
NOK-HI-002	**Reader-phone interface performance** The interface between reader component and the mobile phone shall not significantly limit system performance.		This requirement is no longer valid, listed as <obsolete> in D7.12.
NOK-PHR-001	**Reader component physical dimensions** The reader component shall be small enough to be attached to the mobile phone cover.		This requirement is no longer valid, listed as <obsolete> in D7.12.
NOK-RD-001	**Wireless power transfer from reader to tag** The reader shall be capable of powering the battery-less RF tag (at least the communication functions) with UHF RF signal.		Verified by simulations and link budget calculations, but cannot be tested until fully integrated solution is available (beyond MINAmi scope)
NOK-RD-002	**Powering of active sensors** It shall not be necessary to power up any active sensors attached to the RF tag.		This requirement is no longer valid, listed as <obsolete> in D7.12.
NOK-RD-003	**UHF transmitter** It shall be sufficient to use a separated UHF transmitter attached to the reader (i.e. integrated transmitter is not compulsory)	OK	A separate UHF transmitter is used.
NOK-PR-004	**Data transfer between reader and phone memory** Mobile phone should contain an interface		This requirement is no longer valid, listed as <obsolete> in D7.12.

	fast enough for wired data transfer from the reader hardware to the memory of mobile phone.		

4.1.2 Test Cases tested

The test cases to be tested appear in the document D7.14 *Test & validation plan for the final set of demonstrators.*

Test Case TAG	Description	Status	Comments
TC_T74_01	Tag detection. When a memory tag is in the readers proximity, the reader must detect the presence of the tag and automatically establish a communication link with the tag. The reader device is configured to continuously scan for any possible tag within its reading distance. When a tag is detected, the reader attempts to establish a bi-directional communication link with the tag. If the communication link is working, then the end user is notified that a tag is found. The user may also be prompted to initiate further actions, for example 'initiate data transfer'.	OK	Detection and identification of tag and automatic connection setup between reader and tag works as expected.
TC_T74_02	Data transfer from tag to reader. When the reader has detected that a tag is present in its operating field, the end user can initiate data transfer from tag to reader. QVGA video stream is transferred from tag to reader and the video is displayed on the laptop that is attached to the reader prototype via USB connection.	OK	Video transfer and playback works. Transfer speed is good enough to enable good video quality.

4.2 Bugs Report

There are no bugs to report.

Section 5 - Demonstrator Manuals

5.1 User Manual

This section describes how to use the “memory download from passive tag” demonstrator. Initiating the download is very simple; the user does not have to start any application or press any key, it is enough to simply touch the tag with the mobile phone.

When user touches the memory tag with her mobile phone, the title of the movie clip is displayed on the phone screen, followed by these three options:
- download from tag
- download via internet
- buy tickets

The first option “download from tag” is the default action. If the user touches the tag again, this action is selected automatically. Alternatively, the user can manually select any of the options listed above using the navigation buttons of the phone.

When download from tag (or via internet) is selected, the movie clip stored in the tag is transferred into the mobile phone. After the data transfer is complete, the Media Player application is automatically lauched and the movie clip can be shown on the phone screen.

Note that the “download via internet” option does not require an active internet connection. In both download cases, the movie clip is already pre-installed in the phone and the demo application simply emulates the time that it would take to download the file either from the tag or via internet.

If the user selects the third option “buy tickets”, the phone ask confirmation to create a call to the telephone number that is stored in the tag. By selecting “Yes”, the user can call this number and reserve tickets for the movie.

The demo application can be closed either by pressing the “finish” button or alternatively pressing the red “end call” button.

5.2 Installation Manual

The SW implementation of the “memory denwload from passive tag” demonstrator consists of two separate applications, ***TagWriter*** and ***MINAmITag***. TagWriter application is used to write the necessary configuration data to an NFC tag. MINAmITag is the actual demo application that is automatically launched when the mobile phone detects an NFC tag within its reading range. Both of these applications are implemented as Java MIDLets and they can be installed to the phone (Nokia 6212 NFC) using the Nokia PC suite. For both applications, the installer consists of a single *.jar file (for example, TagWriter.jar).

5.2.1 Initialization of the NFC tag using TagWriter

When TagWriter is started, the application prints two choices: “write” or “read”. Select “write” to enter write mode. Next, the following information can be entered:
- title of the tag (= name of the movie)
- size of the data (movie clip) in megabytes
- location of the movie file (c:/gallery/video/<filename>.3gp)

- telephone number for ticket reservation

The title text should be set to match the movie clip that is used in the demo. Size of the clip can be set to any value and it is used to adjust the transfer delay. (Total transfer delay is calculated based on file size and assumed transfer speed).

The third parameter is the location of the movie. The movie clip should be in *.3gp format and stored in the video clip gallery of the phone.

The final parameter to be written to the tag is the telephone number for ticket reservation. Any number (valid or invalid) can be entered to this field.

When all of the data fields described above have been filled in, the information can be written to an NFC tag by touching the tag with the mobile phone. The data is permanently stored in the tag and this action needs to be done only once for each tag.

5.2.2 Running the MINAmITag application

After the MINAmITag application has been installed, there is no need for any further configurations. The demo application will launch automatically whenever the mobile phone detects an NFC tag that has been programmed using the instructions described in the previous section.

References

[1] MINAmI delivarable D7.1, "Basic requirements from application participants", v1.1

[2] ISO/IEC 18092, "Information technology – Telecommunication and information exchange between systems – Near Field Communication – Interface and Protocol (NFCIP-1)"